REMARKS

Claims 1-20 remain pending in the present application. Reconsideration of the pending claims is earnestly solicited.

I. Response to 35 U.S.C. §102 Rejection

Claims 1-20 stand rejected under 35 U.S.C. §102(e) as allegedly being anticipated by *Grey et al.* (U.S. Patent No. 6,401,220). Applicants respectfully traverse this rejection on the grounds that *Grey et al.* does not disclose each and every element of independent claims 1, 10, and 16.

A. Background

Before discussing the *Grey et al.* reference, it may be necessary to set forth background information that is well-known to one of ordinary skill in the art of computer programming. Of significant importance is the term "data type." First of all, every variable has a data type. "Standard data types" establish that a variable's set of data will have values with predefined characteristics. For instance, if a variable has an "integer" data type, then the variable can only contain integer values. Data types also establish the range of values that a variable can contain. For instance, a "short" integer is limited to a range of number that can be represented by a 16-bit value. In addition to "integer" data types, a variable might also have other data types, such as floating point numbers, characters, strings, and pointers. The list of these "standard data types" depends on availability with respect to the computer language being used. Different computer languages may have different standard data types.

Even though a limited number of such standard data types are built into a language, certain languages allow a programmer to create new data types to meet additional application needs. These data types are typically referred to as "custom data types." If a programmer has a need for a data type that is not effectively represented by the available standard data types, the programmer may create a custom data type to meet these particular needs. In this respect, custom data types will be created with custom characteristics regarding type, size, and range.

B. *Grey et al.*

Grey et al. discloses a system that improves upon the test executive system, but does not improve upon the built-in standard data types or even customized data

types. Instead, *Grey et al.* provides a system that includes "step types." The test executive software of *Grey et al.* expands on the traditional test executive concepts and introduces many new ones. Grey's test executive software includes new concepts and features, including step types, etc. (col. 3, lines 6-11). Grey's system includes various types, including step types, custom data types, and standard data types. A type can thus be a data type or step type (col. 3, lines 35-38). *Grey et al.* does not describe changes to the conventional data types (custom or standard), but instead describes in detail the new concept of "*step types*."

In col. 4, lines 47-64, *Grey et al.* discloses that a sequence comprises a series of steps, wherein a step is typically a test performed on an instrument. In a test sequence with a number of steps, the user oftentimes desires a number of steps that have some commonality of functionality and/or properties. The purpose of the step type is to define common properties and/or operations associated with a plurality of steps, thereby eliminating the need for hard coding the functionality and/or properties in each instance. Furthermore, a step type essentially comprises a custom set of properties and/or operations associated with a step. The step type defines common operations and/or data associated with a test module. Essentially, *Grey et al.* appears to be concerned only with this new concept of "step types" and does not attempt to alter conventional data types. All references to data types in *Grey et al.* are directed to the standard and custom data types that are well known in the art.

C. Claim 1

Independent claim 1 is directed to an ***"active data type"*** for use in a computer program. The active data type comprises an identifier and ***"at least a first algorithm."*** The Office Action seems to allege that *Grey et al.* discloses a data type having an algorithm and refers to col. 6, lines 1-5 and col. 25, lines 31-34 for support. However, it should be noted that the passage in col. 6, lines 1-5 actually refers to a plurality of steps of a test sequence and executing steps of a first step type. *Grey et al.* does not disclose a data type having an algorithm. The passage in col. 25, lines 31-34 refers to storing a type definition when a data type or step type is created, as is well-known. However, *Grey et al.* fails to disclose a data type having an algorithm as claimed.

Furthermore, claim 1 includes that the first algorithm is ***"automatically executed when an attempt is made to access a value associated with the active data type instance."*** Thus, when a value of the active data type is accessed by a read or write operation, the algorithm is executed automatically to process the value according to the algorithm. The Office Action seems to allege that *Grey et al.* discloses such a feature. Applicants disagree with this contention. *Grey et al.* actually refers to automatically storing a type definition and fails to teach or suggest executing an algorithm, associated with a data type, when an attempt is made to access a value associated with the data type. *Grey et al.* fails to disclose an algorithm associated with an active data type and executing the algorithm when an attempt is made to access a value associated with the active data type.

D. Claim 10

Independent claim 10 is directed to an apparatus for executing a computer program, wherein the apparatus comprises logic configured to execute the computer program that utilizes at least one active data type. The ***"active data type"*** comprises an identifier and ***"at least a first algorithm."*** The Office Action seems to allege that *Grey et al.* discloses a data type having an algorithm and refers to col. 6, lines 1-5 and col. 25, lines 31-34 for support. However, it should be noted that the passage in col. 6, lines 1-5 actually refers to a plurality of steps of a test sequence and executing steps of a first step type. *Grey et al.* does not disclose a data type having an algorithm. The passage in col. 25, lines 31-34 refers to storing a type definition when a data type or step type is created, as is well-known. However, *Grey et al.* fails to disclose a data type having an algorithm as claimed.

Furthermore, claim 10 includes that the first algorithm is ***"automatically executed when an attempt is made to access a value associated with the active data type instance."*** Thus, when a value of the active data type is accessed by a read or write operation, the algorithm is executed automatically to process the value according to the algorithm. The Office Action seems to allege that *Grey et al.* discloses such a feature. Applicants disagree with this contention. *Grey et al.* actually refers to automatically storing a type definition and fails to teach or suggest executing an algorithm, associated with a data type, when an attempt is made to access a value associated with the data type. *Grey et al.* fails to disclose an algorithm associated with

an active data type and executing the algorithm when an attempt is made to access a value associated with the active data type.

E. Claim 16

Independent claim 16 is directed to a method for utilizing an ***"active data type"*** in a computer program. The active data type comprises the steps of identifying an instance of the active data type and ***"automatically executing a first algorithm."*** The Office Action seems to allege that *Grey et al.* discloses a data type that executes an algorithm and refers to col. 6, lines 1-5 and col. 25, lines 31-34 for support. However, it should be noted that the passage in col. 6, lines 1-5 actually refers to a plurality of steps of a test sequence and executing steps of a first step type. *Grey et al.* does not disclose a data type having an algorithm. The passage in col. 25, lines 31-34 refers to storing a type definition when a data type or step type is created, as is well-known. However, *Grey et al.* fails to disclose a data type executing an algorithm as claimed.

Furthermore, claim 16 includes that the first algorithm is ***"automatically executed ... when an attempt is made to access a value associated with the active data type instance."*** Thus, when a value of the active data type is accessed by a read or write operation, the algorithm is executed automatically to process the value according to the algorithm. The Office Action seems to allege that *Grey et al.* discloses such a feature. Applicants disagree with this contention. *Grey et al.* actually refers to automatically storing a type definition and fails to teach or suggest executing an algorithm, associated with a data type, when an attempt is made to access a value associated with the data type. *Grey et al.* fails to disclose an algorithm associated with an active data type and executing the algorithm when an attempt is made to access a value associated with the active data type.

F. Claims 2-9, 11-15, and 17-20

Dependent claims 2-9, 11-15, and 17-20 are believed to be allowable for at least the reason that these claims depend from allowable independent claim 1, 10, and 16.

CONCLUSION

In light of the foregoing amendments and for at least the reasons set forth above, Applicant respectfully submits that all rejections have been traversed and that claims 1-20 are in condition for allowance. Favorable reconsideration and allowance of the present application and all pending claims are hereby courteously requested. If, in the opinion of the Examiner, a telephonic conference would expedite the examination of this matter, the Examiner is invited to call the undersigned at (770) 933-9500.

Respectfully submitted,

Glenn W. Brown

Glenn W. Brown
Reg. No. 51,310

THOMAS, KAYDEN,
HORSTEMEYER & RISLEY, L.L.P.
Suite 1750
100 Galleria Parkway N.W.
Atlanta, Georgia 30339
(770) 933-9500

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, postage prepaid, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on 3-10-04

Evelyn Sanders

Signature -